J. Symbolic Computation (2000) **29**, 83–94
doi: 10.1006/jsco.1999.0291
Available online at http://www.idealibrary.com on IDEAL

SFA, a Package on Symmetric Functions Considered as Operators over the Ring of Polynomials for the Computer Algebra System MAPLE

VINCENT PROSPER

Gaspard Monge Institute, University of Marne-la-Vallée, 77454 Marne-la-Vallée Cedex 2, France

Symmetric functions can be considered as operators acting on the ring of polynomials with coefficients in $\mathbb{R}$. We present the package SFA, an implementation of this action for the computer algebra system MAPLE. As an example, we show how to recover different classical expressions of Lagrange inversion, and of Faber polynomials.

1. Introduction

Up to now, two main implementations of the theory of symmetric functions in MAPLE were made in ACE (Ung and Veigneau, 1995) and SF (Stembridge, 1995). The ring of symmetric functions can act in both of them as operators on the ring of polynomials with coefficients in $\mathbb{R}$. Implementations of symmetric functions for systems other than MAPLE are, for instance, SYMMETRICA (Kerber *et al.*, 1992) and SYM for Macsyma (Valibouze, 1989).

In the SFA package — *Symmetric Functions on different Alphabets*, we give the possibility of performing more formal computations by implementing the structure of the Λ-ring of symmetric functions viewed as operators. Symmetric functions are operators on the ring of polynomials. However, one can recover usual symmetric functions (i.e. invariant by permutation): `SfAExpand(e[2](x1+x2+x3))` will produce $\mathtt{x1x2}+\mathtt{x1x3}+\mathtt{x2x3}$, which is indeed the second elementary function in $\{\mathtt{x1},\mathtt{x2},\mathtt{x3}\}$.

The SFA package works together with the ACE (Veigneau, 1998) environment and is also integrated in the $\mu{-}EC$ environment (Lascoux and Prosper, 1998) for MUPAD.

We briefly give some commonly used notations. We recall the notion of a Λ-ring. We present the functionalities of SFA. Finally, we show how to derive easily some formulæ for the Lagrange inversion of formal series, and for Faber polynomials.

2. Notations

Let $\mathbb{N}$ be the set of positive natural integers. Let $\mathbb{N}_* = \mathbb{N}\setminus\{0\}$.

A partition μ is a vector in $\mathbb{N}_*^\ell$, $\ell \geq 0$, whose components, called the parts of μ, are sorted in decreasing order. $|\mu| := \sum_{i=1}^{\ell}\mu_i$ is called the *weight* of μ and $\ell(\mu) := \ell$ is called the *length* of μ.

Let $\mathbb{X} = \{x_1, x_2, \ldots\}$ be a set of variables. Let Λ^i be an operator on $\mathbb{R}[\mathbb{X}]$. Let $\Lambda^\mu(P)$ de-

0747–7171/00/010083 + 12 $35.00/0

note the product $\prod_{i=1}^{\ell} \Lambda^{\mu_i}(P)$, for each $P \in \mathbb{R}[\mathbb{X}]$. This is not a composition of operators, we shall also consider that later.

3. The Structure of a Λ-ring

Let P and Q be two elements of the ring $\mathbb{R}[\mathbb{X}]$ of polynomials with coefficients in $\mathbb{R}$. Let $x \in \mathbb{X}$ and $r \in \mathbb{R}$. Let μ and ρ denote some partitions. One defines the operators Λ^i over $\mathbb{R}[\mathbb{X}]$, $i \in \mathbb{Z}$ by the following relations:

$$\begin{aligned}
&\Lambda^i(P) = 0, \quad i < 0, \qquad \Lambda^0(P) = 1, \qquad \Lambda^1(P) = P,\\
&\Lambda^i(x) = 0, \quad i > 1, \qquad \Lambda^i(r) = \binom{r}{i} = \left(\Pi_{j=1}^{i}(r-j+1)\right)/i!,\\
&\Lambda^i(P+Q) = \sum_j \Lambda^{i-j}(P)\Lambda^j(Q), \qquad (3.1)\\
&\exists\ c_\mu^\rho \in \mathbb{Z} \text{ such that } \Lambda^i(PQ) = \sum_{|\mu|=|\rho|=i} c_\mu^\rho \Lambda^\mu(P)\Lambda^\rho(Q).
\end{aligned}$$

These axioms totally define the $\Lambda^k(P)$ for all $k \in \mathbb{Z}$ and $P \in \mathbb{R}[\mathbb{X}]$ (in fact, the axioms are simpler when given in the basis of the Ψ^i operators — see equation (4.4)). For the definition of general Λ-rings, one has to add an axiom for the composition $P \to \Lambda^j(P) \to \Lambda^i(\Lambda^j(P))$. Here, because we only deal with polynomials, $\Lambda^j(P)$ is a polynomial and thus $\Lambda^i(\Lambda^j(P))$ is determined by the axioms (3.1).

When P is the polynomial $P = x_1 + x_2 + \cdots + x_n$, from (3.1), one obtains that:

$$\Lambda^n(x_1 + x_2 + \cdots + x_n - 1) = \sum (-1)^{n-k}\Lambda^k(P) = \prod (x_i - 1), \qquad (3.2)$$

i.e. $\Lambda^k(P)$ is the elementary symmetric function of the alphabet $\mathbb{X} = \{x_1, x_2, \ldots, x_n\}$ of degree k $(= \sum_{1\le i_1<\cdots<i_k\le n} x_{i_1}\ldots x_{i_k})$.

From the axioms, we deduce for $P = \sum ru$, $r \in \mathbb{R}$ and $u \in \mathbb{X}^*$, (i.e. u monomial):

$$\sum z^i \Lambda^i(P) = \prod (1 + zu)^r, \qquad (3.3)$$

and this could have been taken as a definition of the Λ-ring structure instead of (3.1).

4. Other Bases

Instead of the elementary symmetric functions, one could take any other algebraic basis of the ring of symmetric functions and define accordingly the action of elements of this basis on the ring of polynomials $\mathbb{R}[\mathbb{X}]$.

For instance, complete functions S^k, viewed as operators on $\mathbb{R}[\mathbb{X}]$ enjoy:

$$\begin{aligned}
&S^i(P) = 0, \quad i < 0, \qquad S^0(P) = 1, \qquad S^1(P) = P,\\
&S^i(x) = x^i, \quad i > 1, \qquad S^i(r) = \left(\prod_{j=1}^{i}(r+j-1)\right)/i!,\\
&S^i(P+Q) = \sum_j S^{i-j}(P)S^j(Q), \qquad (4.1)
\end{aligned}$$

$$\exists\ c^\rho_\mu \in \mathbb{Z} \text{ such that } S^i(PQ) = \sum_{|\mu|=|\rho|=i} c^\rho_\mu S^\mu(P) S^\rho(Q).$$

This action is equivalent to the one described by axioms (3.1), because in fact:

$$S^k(P) = (-1)^k \Lambda^k(-P). \tag{4.2}$$

Accordingly, (4.1) becomes, for $P = \sum ru$, $r \in \mathbb{R}$ and $u \in \mathbb{X}^*$:

$$\sum z^i S^i(P) = \prod (1 - zu)^{-r}. \tag{4.3}$$

On the other hand, powers sums Ψ^i provide simpler axioms than elementary symmetric functions or complete functions. Let $P = \sum ru$, $r \in \mathbb{R}$ and $u \in \mathbb{X}^*$, and let $i \geq 1$, then:

$$\Psi^i(P) = \sum ru^i. \tag{4.4}$$

In particular:

$$\begin{aligned} \Psi^i(P+Q) &= \Psi^i(P) + \Psi^i(Q) \\ \Psi^i(PQ) &= \Psi^i(P) \times \Psi^i(Q), \end{aligned} \tag{4.5}$$

and moreover:

$$\Psi^i(\Psi^j(P)) = \Psi^{ij}(P), \tag{4.6}$$

which is also known as the *plethysm* operation. Let us note that $\Psi^i(x_1 + x_2 + \cdots + x_n) = \sum x^i_j$ is indeed the power sum function Ψ^i over the alphabet $\{x_1, x_2, \ldots, x_n\}$.

Two other (linear) bases of symmetric functions may also be considered: the Schur functions and the monomial functions, whose elements are commonly written as S_μ and Ψ_μ, μ any partition. Usually — see (Macdonald, 1995), the Schur functions are defined by:

$$S_\mu := \det |S_{\mu_j} + j|. \tag{4.7}$$

The monomial symmetric function Ψ_μ is defined to be the sum of all monomials of degree μ over a given alphabet. That is to say, $\Psi_\mu(x_1 + x_2 + \cdots + x_\ell)$ is the sum of all distinct monomials image by permutation of $x_1^{\mu_1} \ldots x_\ell^{\mu_\ell}$.

In (Macdonald, 1995), products of elementary functions (Λ^μ) are denoted by $\mathtt{e}_\mu$, products of complete functions (S^μ) are denoted by $\mathtt{h}_\mu$, products of power sum functions (Ψ^μ) are denoted by $\mathtt{p}_\mu$, monomial functions (Ψ^μ) are denoted by $\mathtt{m}_\mu$, and Schur functions (S^μ) are denoted by $\mathtt{s}_\mu$. We also use this syntax in `SFA`.

5. The SFA Package

The `SFA` package, which is already integrated in the last release of the `ACE` environment for Maple (Veigneau, 1998), implements the action of all common symmetric functions on $\mathbb{R}[\mathbb{X}]$. On-line documentation and downloadable software are provided at the URLs (Veigneau, 1998; Lascoux and Prosper, 1998) given at the end of this paper. We detail the functionalities of `SFA`.

5.1. syntax of SFA

symmetric functions

Symmetric functions in `SYMF` (Ung and Veigneau, 1995) or `SF` (Stembridge, 1995) have either the type $\mathtt{b}n$, $n \geq 0$, when `b` is an algebraic basis of the ring of symmetric functions

— $\mathtt{b} \in \{\mathtt{e},\mathtt{h},\mathtt{p}\}$, or $\mathtt{b}[n_1, n_2, \ldots, n_k]$, $n_1 \geq n_2 \geq \cdots n_k \geq 1$, when b is a linear basis of the ring of symmetric functions — $\mathtt{b} \in \{\mathtt{s},\mathtt{m}\}$.

In SFA, all the symmetric functions have the shape $\mathtt{b}[n_1, n_2, \ldots, n_k](A)$, $n_1 \geq n_2 \geq \cdots n_k \geq 1$ — $\mathtt{b} \in \{\mathtt{e},\mathtt{h},\mathtt{p},\mathtt{s},\mathtt{m}\}$ and A is any alphabet (alphabets can be any linear combination of formal alphabets, see below). When b is an algebraic basis, the object $\mathtt{b}[n_1, n_2, \ldots, n_k](A)$ is considered as a product of bis.

For instance, the symmetric function $2(\Lambda^1)^2\Lambda^2 + qS^1S_{[3,2]}$ shall be written down in SFA:

```
2*e[2,1,1](A1) + q*h[1](A1)*s[3,2](A1),
```

A1 being a formal alphabet, contrary to SYMF where it is written:

```
2*e1^2*e2 + q*h1*s[3,2].
```

And we have $\mathtt{e}[\mu](A) = \Lambda^\mu(A)$, $\mathtt{h}[\mu](A) = S^\mu(A)$, $\mathtt{p}[\mu](A) = \Psi^\mu(A)$, and $\mathtt{s}[\mu](A) = S_\mu(A)$, $\mathtt{m}[\mu](A) = \Psi_\mu(A)$.

ALPHABETS

Any formal alphabet in SFA must be written as An, where n is any positive integer. Any other variable is considered as a constant, i.e. an element of $\mathbb{R}$, except symmetric functions, when they are encoded as operators on $\mathbb{R}[\mathbb{X}]$. Hence, B1 is treated as an element of $\mathbb{R}$ in the expression p[3,2](B1) (= B1^2, according to equation (4.4)), contrary to p[3,2](A1), where A1 is recognized as a formal alphabet.

More generally, an alphabet is recursively defined to be either a sum of alphabets $\sum \alpha_i A_i, \alpha_i \in \mathbb{R}$, or a product of alphabets, or a symmetric function of SFA, or a formal alphabet $\mathtt{A}_i$ or a scalar. Any variable $x_i \in \mathbb{X}$ can be encoded by a formal alphabet $\mathtt{A}_i$ or through a special declaration (see Section 5.4).

5.2. FROM SYMF/SF TO SFA

Any symmetric function of SYMF (or SF) on the bases $\mathtt{e},\mathtt{h},\mathtt{p},\mathtt{s},\mathtt{m}$ can be converted into SFA objects through the Sf2SfA function:

```
> Sf2SfA( 2*e1^2*e2 + q*h1*s[3,2] );

                  2 e[2, 1, 1](A1) + q s[3, 2](A1) h[1](A1).
```

One can also specify a particular alphabet:

```
> Sf2SfA( k+s[3,1]^2, A2+2*A3 );

                                              2
                           k + s[3, 1](A2 + 2 A3).
```

5.3. FROM SFA TO SYMF/SF

Any symmetric function in SFA can be converted into its equivalent in SYMF (or SF), via the SfA2Sf call:

```
> SfA2Sf( e[2](A1)*s[3,1](A1+A2) + k );

                          s[3, 1] e2 + k.
```

If we only want symmetric functions on particular alphabets to be treated, we must specify the list of alphabets:

```
> SfA2Sf( e[2](A1)*s[3,1](A1+A2) + k*p[2,1](A2)*m[2](A1+A2),  [A2, A1+A2] );

                  s[3, 1] e[2](A1) + p2 p1 m[2] k.
```

5.4. ACTION OF A SYMMETRIC FUNCTION OVER $\mathbb{R}[\mathbb{X}]$

A symmetric function of SFA, when its alphabet does not have the shape An, can be expanded — see equations (3.1, 4.1, 4.4). The SfAExpand function realizes this operation:

```
> SfAExpand( p[3,1,1](3) );

                                27
```

or:

```
> SfAExpand( p[1](A1-A2) * h[1,1](2*A1+k)^2 );

    16 p[1](A1) h[1, 1, 1, 1](A1) + 32 p[1](A1) k h[1, 1, 1](A1)

                         2                                    3          4
         + 24 p[1](A1) k  h[1, 1](A1) + 8 p[1](A1) h[1](A1) k  + p[1](A1) k

         - 16 p[1](A2) h[1, 1, 1, 1](A1) - 32 p[1](A2) k h[1, 1, 1](A1)

                         2                                    3          4
         - 24 p[1](A2) k  h[1, 1](A1) - 8 p[1](A2) h[1](A1) k  - p[1](A2) k.
```

One can compute more intricate examples, for instance, the plethysm operation:

```
> SfAExpand( m[2,1](k*A2 - p[3](2*A1)) );

   4 p[6, 3](A1) + 2 p[9](A1) - 2 k p[1](A2) p[6](A1) - 2 k p[2](A2) p[3](A1)

           2
        + k  p[2, 1](A2) - k p[3](A2).
```

For all symmetric functions in SFA, the alphabet may contain some variables (which are indeed alphabets of cardinal 1). First, the user must specify the names of the variables:

```
> SfAVars({ y1, y2, {x} }):
```

means that y1, y2 and all the xis are now considered as variables. Thus:

```
> SfAExpand(p[3](x1+x2+x3) + q*e[2,1](y1+y2) + p[2](A2));

                   3      3      3        2              2
                 x1  + x2  + x3  + q y1  y2 + q y1 y2  + p[2](A2).
```

It is of course possible to do formal computations with alphabets A1, A2,..., and then to specify their variables and to expand the result as shown above.

5.5. PUTTING AN ELEMENT OF SFA INTO A SPECIAL BASIS

As in SYMF or SF, any element of SFA can be expanded on any recognized basis $(\mathtt{e}, \mathtt{h}, \mathtt{p}, \mathtt{s}, \mathtt{m})$. The result may thereafter be expanded via SfAExpand. The available functions are: ToeA, TohA, TopA, TosA and TomA.

Let us take an example:

```
> TomA( s[2,1](k*A1)*p[1,1](A1) );

      m[2, 1](k A1) m[2](A1) + 2 m[2, 1](k A1) m[1, 1](A1)

           + 2 m[1, 1, 1](k A1) m[2](A1) + 4 m[1, 1, 1](k A1) m[1, 1](A1)

> SfAExpand(");
   - 1/3 m[2](A1) k p[3](A1) - 2/3 m[1, 1](A1) k p[3](A1)

                           3                                    3
        + 1/3 m[2](A1) k  p[1, 1, 1](A1) + 2/3 m[1, 1](A1) k  p[1, 1, 1](A1).
```

The indexing vectors of the symmetric functions do not need to be partitions; they are reordered according to the corresponding rule for the basis in cause (see (Macdonald, 1995)):

$$\begin{aligned}
&\mathtt{s}[\ldots,\mathtt{i},\mathtt{j},\ldots](\mathrm{P}) = -\mathtt{s}[\ldots,\mathtt{j}-1,\mathtt{i}+1,\ldots](\mathrm{P}), \qquad \text{if } \mathtt{i}<\mathtt{j},\\
&b[\ldots,\mathtt{i},\mathtt{j},\ldots](\mathrm{P}) = b[\ldots,\mathtt{j},\mathtt{i},\ldots](\mathrm{P}), \qquad \text{if } \mathtt{i}<\mathtt{j}, b\neq \mathtt{s}.
\end{aligned} \tag{5.1}$$

```
> TosA( s[1,4](A1) + e[2,3](A2) );

   - s[3, 2](A1) + s[1, 1, 1, 1, 1](A2) + s[2, 1, 1, 1](A2) + s[2, 2, 1](A2).
```

One can also choose to apply a change of basis on specific alphabets:

```
> TopA( s[2](A1) + e[2,3](A2)*m[2,1](q*A3), [A1, q*A3]  );

          1/2 p[2](A1) + 1/2 p[1, 1](A1) + e[2, 3](A2) p[2, 1](q A3)

               - e[2, 3](A2) p[3](q A3).
```

5.6. FROM THE BASIS OF MONOMIALS TO SFA

When analyzing results of computations on polynomials, one may want to identify symmetric functions. This is achieved through the Pol2SfA function.

Suppose that we want to recognize a symmetric function in the polynomial:

$$\begin{aligned}\mathtt{P} := {} & x_3^3x_1 + x_3^3x_2 + x_3^2x_2^2 + 2x_1x_2x_3^2 + x_3^2x_1^2 + x_3x_2^3 \\ & +2x_3x_1x_2^2 + 2x_3x_1^2x_2 + x_3x_1^3 + x_1x_2^3 + x_1^2x_2^2 + x_1^3x_2 \\ & +qy_1^2y_2 + qy_1y_2^2 + A2,\end{aligned}$$

```
> Pol2SfA( Pol2SfA(P, [x1,x2,x3]), [y1,y2] );

  2 m[2, 1, 1](x1 + x2 + x3) + m[2, 2](x1 + x2 + x3) + m[3, 1](x1 + x2 + x3)

       + q m[2, 1](y1 + y2) + A2

> PP := ToeA( TosA(", [x1+x2+x3]), [y1+y2] );

        PP := - 3 s[1, 1, 1, 1](x1 + x2 + x3) + s[3, 1](x1 + x2 + x3)

             + q e[2, 1](y1 + y2) - 3 q e[3](y1 + y2) + A2.
```

The result PP is the right one, but it can be simplified, since for example, `e[3](y1 + y2)` is null:

```
> SfAVars({ {x}, {y} }):
> map(t -> if SfAExpand(t)=0 then 0 else t fi, PP);

                s[3, 1](x1 + x2 + x3) + q e[2, 1](y1 + y2) + A2.
```

5.7. the Ω-automorphism

The Ω-automorphism is an involution defined on symmetric functions by:

$$\begin{aligned}&\mathtt{h}[\mu](A) \leftrightarrow \mathtt{e}[\mu](A)\\ &\mathtt{s}[\mu](A) \leftrightarrow \mathtt{s}[\mu^{\sim}](A)\\ &\mathtt{p}[\mu](A) \leftrightarrow -1^{\ell(\mu)+|\mu|}\mathtt{p}[\mu](A),\end{aligned}$$

where $\mu^{\sim}$ is the conjugate partition of μ.

```
> SfAOmega( s[3,2](A1+A2) - p[2,1](A1) );

                  s[2, 2, 1](A1 + A2) + p[2, 1](A1).
```

Here, one can also apply the `SfAOmega` function on symmetric functions over specific alphabets.

5.8. decomposing an element of SFA

One can apply any transformation on each element of an object of SFA through its decomposition. The `SfA2TableVar` function returns all symmetric functions appearing in a polynomial expression:

```
> tt := SfA2TableVar( m[2](A1) - q*e[3,2](A2)*p[1](A1) + e[5](A1) - e[3](A1) );

              tt := table([   e = table([ A2 = {[3, 2]}, A1 = {[3], [5]} ])
                              h =
                              s =
                              p = table([ A1 = {[1]} ])
                              m = table([ A1 = {[2]} ])       ])

> tt[e][A2];

                                   {[3, 2]}.
```

In order to preserve the structure of the former function, the user can convert it into a function of the SYMF package (Veigneau, 1998), and then call the Sf2Table function of SYMF, which builds a table indexed by partitions, representing the element of the symmetric function algebra in a given basis. He or she applies the desired transformation and comes back to SFA via Sf2SfA.

6. Cauchy Formula

The action of S^is on a product of polynomials was stated with the help of universal coefficients in axioms (4.1). In fact, in the basis of Schur functions, the expansion of $S^i(PQ)$ is given explicitly by the Cauchy formula:

$$S^i(PQ) = \sum_{|\mu|=i} S_\mu(P)S_\mu(Q), \tag{6.1}$$

where the Schur functions $S_\mu(P)$ are defined by equality (4.7).

Moreover, one has also the expansion of $S^i(PQ)$ in any pair of dual bases:

$$\begin{aligned} S^i(PQ) &= \sum_{|\mu|=i} S_\mu(P)S_\mu(Q) \\ &= \sum_{|\mu|=i} \Psi_\mu(P)S^\mu(Q) \\ &= \sum_{|\mu|=i} S^\mu(P)\Psi_\mu(Q) \\ &= (-1)^i \sum_{|\mu|=i} \Lambda^\mu(-P)\Psi_\mu(Q) \\ &= (-1)^i \sum_{|\mu|=i} \Psi_\mu(P)\Lambda^\mu(-Q) \\ &= \sum_{|\mu|} (\Psi^\mu(P)\Psi^\mu(Q))/\langle \Psi^\mu, \Psi^\mu \rangle, \end{aligned} \tag{6.2}$$

where $\langle\ ,\ \rangle$ is the usual scalar product on symmetric functions. Cauchy formula is implemented in its different forms in SFA.

7. Application: Lagrange Inversion

The Lagrange inversion problem is: given a formal series $f = x + \cdots$, one looks for its inverse, i.e. for g such that $g(f(x)) = x$. Following Lagrange, several authors have explicited different expressions of the coefficients of g, or its powers. All these expressions can be written in terms of Λ-rings as follows.

Let us introduce an alphabet A, and write $f = x\sum_{n=0}^{\infty}(-x)^n S_n(A)$. Then the classical formulæ can all be written in a compact manner:

$$g^p(x) = p\ x^p \sum_{i\geq 0} x^i \Lambda^i\big((i+p)A\big)/(i+p). \tag{7.1}$$

The coefficient $\Lambda^i\big((i+p)A\big)/(i+p)$ can be expanded in many different ways due to the Cauchy formula (6.1) and thus provides as many various expressions of Lagrange inversion formula that one can recognize in classical literature:

— in the product of complete functions basis: $\frac{(-1)^i}{i+p}\sum_{|\mu|=i}\Psi_\mu(-(i+p))S^\mu(A)$;
— in the product of elementary functions basis: $\frac{1}{i+p}\sum_{|\mu|=i}\Psi_\mu(i+p)\Lambda^\mu(A)$;
— in the Schur function basis: $\frac{1}{i+p}\sum_{|\mu|=i}S_\mu(i+p)S_{\mu^\sim}(A)$;
— in the monomial function basis: $\frac{1}{i+p}\sum_{|\mu|=i}\Lambda^\mu(i+p)\Psi_\mu(A)$
$= \frac{1}{i+p}\sum\prod_{j=1}^{\ell(\mu)}\binom{\mu_j}{i+p}\Psi_\mu(A)$;
— in the basis of products of power sums: $\frac{(-1)^i}{i+p}\sum_{|\mu|=i}\frac{\Psi^\mu(-(i+p))\Psi^\mu(A)}{\langle\Psi^\mu,\Psi^\mu\rangle}$.

The fact that Lagrange inversion is an involution on formal series is not straightforward on expression (7.1), see (Lenart, 1998).

All the coefficients are directly computable with SFA, starting from (7.1). For instance, the Ostrowski formula is:

$$g^p(x) = p\ x^p \sum_{\mu} x^{|\mu|}\frac{\Psi_\mu\big(-(|\mu|+p)\big)f^\mu}{|\mu|+p}, \tag{7.2}$$

where f^μ is the product $f_{\mu_1}f_{\mu_2}\cdots$, with f_{μ_i} the coefficient of x^i in f.
Let us translate Lagrange coefficients with SFA:

```
> LagCoeff := (i,p) -> p/(i+p) * e[i]( (i+p)*A1 ):


> sum('x^(i+1) * TohA( SfAExpand(LagCoeff(i,1)) )', 'i'=0..5);

     2              3
x + x  h[1](A1) + x  (- h[2](A1) + 2 h[1, 1](A1))

       4                                                          5
     + x  (h[3](A1) - 5 h[2, 1](A1) + 5 h[1, 1, 1](A1)) + x  (- h[4](A1)

     + 6 h[3, 1](A1) + 3 h[2, 2](A1) - 21 h[2, 1, 1](A1) + 14 h[1, 1, 1, 1](A1)

        6
    ) + x  (28 h[3, 1, 1](A1) - 7 h[3, 2](A1) + h[5](A1) - 84 h[2, 1, 1, 1](A1)

     + 42 h[1, 1, 1, 1, 1](A1) + 28 h[2, 2, 1](A1) - 7 h[4, 1](A1)).
```

More concretely, for $p = 1$, one has:

$$
\begin{aligned}
g(x) = {} & x + x^2 f_1 + x^3(-f_2 + 2f_1f_1) + x^4(f_3 - 5f_2f_1 + 5f_1f_1f_1) \\
& + x^5(-f_4 + 6f_3f_1 + 3f_2f_2 - 21f_2f_1f_1 + 14f_1f_1f_1f_1) \\
& + x^6(28f_3f_1f_1 - 7f_3f_2 + f_5 - 84f_2f_1f_1f_1 + 42f_1f_1f_1f_1f_1 + 28f_2f_2f_1 - 7f_4f_1) + \cdots
\end{aligned}
$$

as stated by Ostrowski in equation (7.2).

8. Application: Faber Polynomials

A problem similar to Lagrange inversion is: let A be an alphabet and n an integer. Let $\lambda_z(A) = \sum_{i \geq 0} z^i \Lambda^i(A)$, then there exist a unique polynomial $F_n(x) = x^n + \cdots$ of degree n, called a *Faber polynomial*, such that:

$$F_n\left(\frac{\lambda_z(A)}{z}\right) = z^{-n} + z(\ldots), \tag{8.1}$$

where the coefficient $(\ldots)$ is a formal series in positive powers of z. Indeed, equation (8.1) is a system of n equations (nullity of the coefficients of $z^{-n+1} \ldots z^0$) and determines uniquely the n unknown coefficients of $F_n(x)$ (*f.* (Schur, 1945; Jabotinsky, 1953; Lavoie and Tremblay, 1981)).

Proposition 8.1. *The Faber polynomials $F_n(z)$ are equal to:*

$$F_n(z) = (-1)^n \Psi_n(A) + \sum_{i=1}^{n} \frac{n}{i} \Lambda^{n-i}(-iA)\ z^i. \tag{8.2}$$

Sketch of Proof. We must verify that in the expansion of:

$$(-1)^n \Psi_n(A) + \sum_{i=1}^{n} \frac{n}{i} \Lambda^{n-i}(-iA) \frac{\lambda_z(iA)}{z^i}, \tag{8.3}$$

the coefficients of $z^{-n+1}, \ldots, z^{-1}, z^0$ vanish, or equivalently, that the coefficients of $z, \ldots,$ z^{n-1}, z^n are null in the expression:

$$(-z)^n \Psi_n(A) + \sum_{i=1}^{n} \frac{n}{i} \Lambda^{n-i}(-izA) \lambda_z(iA). \tag{8.4}$$

Now these identities can be checked by derivation of generating series as in the case of Lagrange inversion.□
Let us implement formula (8.4):

```
SfAVars( {z} );              # z is a variable when appearing in alphabets...

Lambda := proc(p,k) local i;           # lambda_z(p A),  k+1 first terms...
  1 + sum( 'z^i * e[i](p*A1)', 'i=1..k' );
end;

EvalFaber := proc(n,k) local i;     # F_n( lambda_z(A) ) modulo z^k, k>n...
  (-z)^n * p[n](A1) +
  sum( 'expand( n/i * e[subs(0=NULL, n-i)]( -i*z*A1 ) * Lambda(i, k-n+i))',
       'i=1..n' );
```

```
end;
```

Now we compute the coefficients of $z, \ldots, z^{n-1}, z^n$ in equation (8.4):

```
Coeffs := proc(sfa) local r, l, i;              # Table of coefficients...
  l := [ coeffs(collect(SfAExpand(sfa),z), z, 'r') ];
  map(SfAExpand, map(ToeA, table([ seq(op(i, [r])=l[i], i=1..nops(l)) ])));
end;
```

Let us verify the nullity of the coefficients for $n = 2, 3, 4$:

```
> seq( Coeffs(EvalFaber(i,i+1)), i=2..4 );
```

```
   table([               , table([              , table([
       1  = 1                 1  = 1                 1  = 1
        3                      3                      3
       z  = 2 e[3](A1)        z  = 0                 z  = 0
       z  = 0                  4                      4
        2                     z  = 3 e[4](A1)        z  = 0
       z  = 0                 z  = 0                 z  = 0
   ])                          2                      5
                              z  = 0                 z  = 4 e[5](A1)
                          ])                          2
                                                     z  = 0
                                                 ])
```

But the system gives more coefficients, for instance the coefficient of z^5 in $F_3(\lambda_z(A)/z)$ is $3\mathtt{e}[6,2](A) + 3\mathtt{e}[4,2,2](A) + 3\mathtt{e}[3,3,2](A) + 3\mathtt{e}[8](A) + 6\mathtt{e}[5,3](A) + 3\mathtt{e}[4,4](A)$. Recall that $\Lambda^\mu(A)$ is denoted in `SFA` by $\mathtt{e}[\mu](A)$. In fact, extensive experimental computations lead to the following conjecture.

Conjecture 8.1. *Faber polynomials, evaluated at $\lambda_z(A)/z$ are nonnegative in the basis of products of elementary symmetric functions of A, that is:*

$$F_n\left(\frac{\lambda_z(A)}{z}\right) = z^{-n} + \sum_{\mu} z^{|\mu|-n} C_\mu \Lambda^\mu(A), \tag{8.5}$$

with C_μ positive, the sum being on all partitions of length $\leq n$.

References

Jabotinsky, E. (1953). Representation of functions by matrices. Application to Faber polynomials. In *Proceedings of the American Mathematical Society*, volume 4, pp. 546–553.

Kerber, A., Kohnert, A., Lascoux, A. (1992). SYMMETRICA, an object oriented computer algebra system for the symmetric group. *J. Symb. Comput.*, **14**, 195–203.

Lascoux, A., Prosper, V. (1998). μ–EC, a MuPAD Environment for Combinatorics, Available online at `http://phalanstere.univ-mlv.fr/~muec`.

Lenart, C. (1998). Lagrange Inversion and Schur Functions, Preprint MIT.

Lavoie, J. L., Tremblay, R. (1981). The Jabotinski matrix of a power series. *SIAM J. Math. Anal.*, **12**, 819–825.

Macdonald, I. G. (1995). *Symmetric Functions and Hall Polynomials*, 2nd Edition, Oxford Science Publications.

Schur, I. (1945). On Faber polynomials. *Am. J. Math.*, **67**, 33–41.

Stembridge, J. (1995). SF, a maple package for symmetric functions. *J. Symb. Comput.*, **20**, 755–768.

Thibon, J.-Y., Veigneau, S. (1996). Using ACE, an algebraic combinatorics environment for MAPLE. In *Proceedings of the Eighth Conference "Formal Power Series and Algebraic Combinatorics"*, pp. 461–468. University of Minnesota.

Ung, B. C. V., Veigneau, S. (1995). ACE, un environnement de calcul en Combinatoire Algébrique. In *Proceedings of the Seventh Conference "Formal Power Series and Algebraic Combinatorics"*, pp. 557–562. Université de Marne-la-Vallée.

Valibouze, A. (1989). Symbolic computation with symmetric polynomials, an extension to Macsyma. In *Conference Computers and Mathematics (1989, MIT, Cambridge, MA)*, pp. 308–320. Springer-Verlag.

Veigneau, S. (1998). ACE, an Algebraic Combinatorics Environment for MAPLE, Available online at `http://phalanstere.univ-mlv.fr/~ace`.

Originally Received 25 July 1998
Accepted 28 May 1999